



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/798,474	03/10/2004	Mark Vincent Scardina	50277-2389	7416
42425 7590 05/31/2007 HICKMAN PALERMO TRUONG & BECKER/ORACLE 2055 GATEWAY PLACE SUITE 550 SAN JOSE, CA 95110-1089			EXAMINER TRAN, QUOC A	
			ART UNIT 2176	PAPER NUMBER
			MAIL DATE 05/31/2007	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/798,474	Applicant(s) SCARDINA ET AL.	
	Examiner Tran A. Quoc	Art Unit 2176	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 29 November 2006.
 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-47 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) ☐ Claim(s) _____ is/are allowed.
 6) ☒ Claim(s) 1-47 is/are rejected.
 7) ☐ Claim(s) _____ is/are objected to.
 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

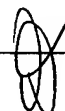
Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) ☐ All b) ☐ Some * c) ☐ None of:
 1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |



Art Unit: 2176

DETAILED ACTION

1. This is a non-final rejection in response to the Amendment/Remarks filed on 11-29-2006.
2. Claims 1-47 are pending and rejected in this action.
3. Effective filing date 03-10-2004, priority date 09-04-2003.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. Claims 1-47 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fry et al. US 20030163603A1 filed 11-26-2002 (hereinafter Fry), in view of Chen et al. US 20030018666A1 filed 07-17-2001 (hereinafter Chen).

Regarding independent claim 1, Fry teaches:

A method comprising the computer-implemented steps of:

while an XML processor performs a validation operation on an XML-based input stream,

(See Fry para 27, discloses an input XML schema 100 , when unmarshalling (unmarshalling converts from a stream to an object- see Fry para 14), binding can be used by feeding an XML stream for an instance of an XML document to generated classes.

Art Unit: 2176

Also, see Fry para 23, discloses validating parser can validate an XML document against an XSD schema and can report any discrepancies. Using the broadest reasonable interpretation, the examiner reads the claimed an **XML processor** as equivalent to validating parser can validate an XML document against an XSD schema and can report any discrepancies as taught by Fry.

In addition, Fry does not explicitly teach, but Chen teaches:

while validating a particular XML element in said XML-based input stream, causing said XML processor to generate one or more messages that indicate how to process specific elements in said XML based input stream based on said particular XML element other than validating said particular XML element, by identifying one or more annotations that are associated with said specific elements particular XML element;

(See Chen fig. 8a-b, shows an intermediate data stream in name/value pair format.)

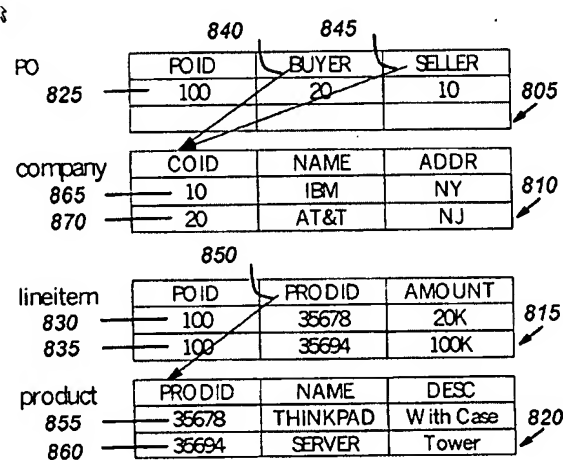


FIG. 8A

Art Unit: 2176

Also, see Chen para 67, teaching the three preparation stages for retrieval from a database in accordance with the preferred embodiment. The first stage 310 populates the EDI structure map or information into relational database tables. The second stage 320 creates annotations that can map the real data (from multiple data sources) to an intermediate data stream (can be a list of name/value pairs or an intermediate XML document), one mapping for each business document type. The third stage 330 annotates the target interoperable DTD that is fixed for all of the different business document types.

**and responding to a request for information about said particular
XML element by providing said one or more messages.**

(See Chen fig. 5b and para 70, discloses the lists of sample EDI validation table that is also stored in a database table 425. The ELEMENT ID column at 520 matches those rows or records in the EDI map table, which have the same value in the CODE column at 502, whose CHOICE column at 506 has a value "element", and whose TYPE column at 508 has a value "ID". The CODE column at 522 in the validation table shows the shorthand terms that can appear as data contents. The VALUE column at 524 shows the long descriptions. For example, in FIG. 5A, the first element of the N1 segment (515) has a code value "98" (516), choice value "element" (517), and type value "ID" (518). It matches the four entries at 530, 531, 532, and 533 that have the same ELEMENT ID "98" in FIG. 5B. The four possible CODE values are "BT" (540), "BY" (541), "SE" (542), and "ST" (543), representing different roles or parties, "Bill-To-Party" (550), "Buying Party" (551), "Selling Party" (552), and "Ship To" (553).

Art Unit: 2176

520 ELEMENT ID	522 CODE	524 VALUE
	143	810 X12 Invoice
	143	850 X12 Purchase Order
530	98	BT Bill-To-Party
		540 550
531	98	BY Buying Party
		541 551
532	98	SE Selling Party
		542 552
533	98	ST Ship To
		543 553
	66	1 D-U-N-S Number, Dun & Bradstreet
	66	8 Vendor UPC Code
	66	9 D-U-N-S+4, D-U-N-S Number with 4 Character Suffix
	66	92 Assigned by Buyer or Buyer's Agent

FIG. 5B

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Fry's XML validating parser, to include a means of validating a particular XML element in said XML-based input stream, causing said XML processor to generate one or more messages that indicate how to process specific elements in said XML based input stream based on said particular XML element other than validating said particular XML element, by identifying one or more annotations that are associated with said specific elements particular XML element and responding to a request for information about said particular XML element by providing said one or more messages as taught by Chen. One of ordinary skill in the art would have been motivated to perform such a modification, because Fry and Chen are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream

Art Unit: 2176

and annotation schema, and allowed for automatic retrieval of data from a relational database into an XML document using an annotated Document Type Definition ("DTD"), and would be usable by other entities (See Chen para 15).

Regarding **independent claim 13**, Fry teaches:

**A method comprising the computer-implemented steps of:
while performing a validation operation on an XML-based input
stream,**

(See Fry para 27, discloses an input XML schema 100, when unmarshalling (unmarshalling converts from a stream to an object- see Fry para 14), binding can be used by feeding an XML stream for an instance of an XML document to generated classes.

Also, see Fry para 23, discloses validating parser can validate an XML document against an XSD schema and can report any discrepancies.

In addition, Fry does not explicitly teach, but Chen teaches:

**receiving a request for information about the state of said
validation operation; and responding to said request by providing
said information about said state of said validation operation;**

(See Chen fig. 4a-4b, 5 and para 68, shows the procedure populates all of the EDI (Electronic Data Interchange) structures into two database tables, and records dictionaries for the element tables 406 and validation tables 408, as shown in FIG. 5B.

Also, see Chen para 21, discloses determined whether the local system is configured to accept deposit to all request fields.

Art Unit: 2176

Also, see Chen para 102, discloses meta data check: issue error if primary key or any non-null column does not have a value. The process may abort with error messages, or accept default (or user provided) values for all the empty fields.)

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Fry's XML validating parser, to include a means of receiving a request for information about the state of said validation operation; and responding to said request by providing said information about said state of said validation operation as taught by Chen. One of ordinary skill in the art would have been motivated to perform such a modification, because Fry and Chen are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and allowed for automatic retrieval of data from a relational database into an XML document using an annotated Document Type Definition ("DTD"), and would be usable by other entities (See Chen para 15).

Regarding **independent claim 39**, Fry teaches:

**A computer-readable medium storing instructions for: a
validator that validates elements and attributes in an XML-based
input stream against information that dictates the structure of
corresponding elements and attributes,**

(See Fry para 27, discloses an input XML schema 100 , when unmarshalling (unmarshalling converts from a stream to an object- see Fry para 14), binding can be used by feeding an XML stream for an instance of an XML document to generated classes.

Also, see Fry para 23, discloses validating parser can validate an XML document against an XSD schema and can report any discrepancies.

Also, see Fry para 35, discloses a schema parser creates a SOM (schema object model) when given an XML schema. Using this SOM, Java classes can be generated that correspond to elements and complex types in the schema. The mapping in both directions can be entered into a type-mapping directory, from complex types to Java classes. The generated classes can implement the DOM interfaces and generic XSD object interfaces, and may not contain any parsing code.)

In addition, Fry does not explicitly teach, but Chen teaches:

**said validator comprising a state machine that responds to
requests for information about validating a first element in said XML-
based input stream, while validating said first element;**

(See Chen fig. 5b and para 70, discloses the lists of sample EDI validation table that is also stored in a database table 425. The ELEMENT ID column at 520 matches those rows or records in the EDI map table, which have the same value in the CODE column at 502, whose CHOICE column at 506 has a value "element", and whose TYPE column at 508 has a value "ID". The CODE column at 522 in the validation table shows the shorthand terms that can appear as data contents. The VALUE column at 524 shows the long descriptions. For example, in FIG. 5A, the first element of the N1 segment (515) has a code value "98" (516), choice value "element" (517), and type value "ID" (518). It matches the four entries at 530, 531, 532, and 533 that have the same ELEMENT ID "98" in FIG. 5B. The four possible CODE values are "BT" (540), "BY" (541), "SE" (542), and

Art Unit: 2176

"ST" (543), representing different roles or parties, "Bill-To-Party" (550), "Buying Party" (551), "Selling Party" (552), and "Ship To" (553).

520 ELEMENT ID	522 CODE	524 VALUE
143	810	X12 Invoice
143	850	X12 Purchase Order
530 98	BT	Bill-To-Party 540 550
531 98	BY	Buying Party 541 551
532 98	SE	Selling Party 542 552
533 98	ST	Ship To 543 553
66	1	D-U-N-S Number, Dun & Bradstreet
66	8	Vendor UPC Code
66	9	D-U-N-S+4, D-U-N-S Number with 4 Character Suffix
66	92	Assigned by Buyer or Buyer's Agent

FIG. 5B

Also, see Chen para 102, discloses Meta data check: issue error if primary key or any non-null column does not have a value. The process may abort with error messages, or accept default (or user provided) values for all the empty fields. Using the broadest reasonable interpretation, the examiner equates the claimed **validator comprising a state machine** as equivalent to EDI validation table and Meta data check: issue error if primary key or any non-null column does not have a value. The process may abort with error messages, or accept default (or user provided) values for all the empty fields as taught by Chen.

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Fry's XML validating parser, to include a means of said validator comprising a state machine that responds to requests for information about validating a first element in said XML-based input stream, while validating said first element as taught by Chen. One of ordinary skill in the art would have been motivated to perform such a modification, because Fry and Chen are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and allowed for automatic retrieval of data from a relational database into an XML document using an annotated Document Type Definition ("DTD"), and would be usable by other entities (See Chen para 15).

Regarding **claims 2-5**, Fry teaches:

wherein the step of receiving a request includes receiving a request via an application program interface through which information about said validation operation can be requested by an external application.

(see Fry para 23, discloses web services, XML binding, Web Service Definition Language (WSDL) code from an existing remote Java interface. A schema can be generated for complex Java data types and can also be mapped to XML Schema Definition language (XSD) types. XSD is an XML-based grammar that can be used to describe the structure of an XML document. A schema-aware validating parser can validate an XML document against an XSD schema and can report any discrepancies.

Also, see Fry para 9-11, describes SOAP (Simple Object Access Protocol) ,
SOAP is a lightweight XML protocol, which can provide both synchronous and
asynchronous mechanisms for sending requests between applications.)

Fry does not explicitly teach, but Chen teaches:

**receiving a request for said one or more annotations; wherein
the step of causing said XML processor to generate one or more
messages is performed in response to said request; wherein the step of
receiving a request includes receiving a request via an application
program interface through which information about said validation
operation can be requested by an external application; to generate one
or more messages that are transmitted in an output stream; and
generate one or more messages before completion of said validation
operation on said XML-based input stream**

(See Chen fig. 5b and para 70, discloses the lists of sample EDI validation table that is also stored in a database table 425. The ELEMENT ID column at 520 matches those rows or records in the EDI map table, which have the same value in the CODE column at 502, whose CHOICE column at 506 has a value "element", and whose TYPE column at 508 has a value "ID". The CODE column at 522 in the validation table shows the shorthand terms that can appear as data contents. The VALUE column at 524 shows the long descriptions. For example, in FIG. 5A, the first element of the N1 segment (515) has a code value "98" (516), choice value "element" (517), and type value "ID" (518). It matches the four entries at 530, 531, 532, and 533 that have the same ELEMENT ID "98"

in FIG. 5B. The four possible CODE values are "BT" (540), "BY" (541), "SE" (542), and "ST" (543), representing different roles or parties, "Bill-To-Party" (550), "Buying Party" (551), "Selling Party" (552), and "Ship To" (553).

Also, see Fry para 23, discloses validating parser can validate an XML document against an XSD schema and can report any discrepancies.

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Fry's XML validating parser, to include a means of receiving a request for said one or more annotations; wherein the step of causing said XML processor to generate one or more messages is performed in response to said request as taught by Chen. One of ordinary skill in the art would have been motivated to perform such a modification, because Fry and Chen are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and allowed for automatic retrieval of data from a relational database into an XML document using an annotated Document Type Definition ("DTD"), and would be usable by other entities (See Chen para 15).

Regarding **claim 6**, the rejection of claim 1 is fully incorporated.

In addition, Fry does not explicitly teach, but Chen teaches:

wherein the step of causing said XML processor to generate one or more messages includes causing said XML processor to generate one or more messages that indicate how to process said particular XML element, only if said particular XML element is

determined valid based on said validation operation on said particular XML element.

(See Chen fig. 5b and para 70, discloses the lists of sample EDI validation table that is also stored in a database table 425. The ELEMENT ID column at 520 matches those rows or records in the EDI map table, which have the same value in the CODE column at 502, whose CHOICE column at 506 has a value "element", and whose TYPE column at 508 has a value "ID". The CODE column at 522 in the validation table shows the shorthand terms that can appear as data contents. The VALUE column at 524 shows the long descriptions. For example, in FIG. 5A, the first element of the N1 segment (515) has a code value "98" (516), choice value "element" (517), and type value "ID" (518). It matches the four entries at 530, 531, 532, and 533 that have the same ELEMENT ID "98" in FIG. 5B. The four possible CODE values are "BT" (540), "BY" (541), "SE" (542), and "ST" (543), representing different roles or parties, "Bill-To-Party" (550), "Buying Party" (551), "Selling Party" (552), and "Ship To" (553).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Fry's XML validating parser, to include a means of causing said XML processor to generate one or more messages includes causing said XML processor to generate one or more messages that indicate how to process said particular XML element, only if said particular XML element is determined valid based on said validation operation on said particular XML element as taught by Chen. One of ordinary skill in the art would have been motivated to perform such a modification, because Fry and Chen are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and

Art Unit: 2176

allowed for automatic retrieval of data from a relational database into an XML document using an annotated Document Type Definition ("DTD"), and would be usable by other entities (See Chen para 15).

Regarding **claims 7-12 respectively:**

are directed to a computer-readable medium carrying one or more sequences of instructions, when executed by one or more processors, causes the one or more processors to perform the method recited in Claims 1-6 respectively and are similarly rejected along the same rationale.

Regarding **claim 14**, Fry teaches:

XML-based input stream is defined in corresponding information that dictates the structure of XML data.

(See Fry para 23, discloses given namespace and element name user can get the appropriate Java classes at runtime in a mode such as generated mode. Given the stream and class, a user can get the populated java instance. The user can also have control over the deserialization. For a Java to XML case, the primitives can be mapped to XSD types. The user can also specify the XSD type. The system can convert the java instance to an XML stream or to a tree, such as a DOM tree.

Art Unit: 2176

In addition, Fry does not explicitly teach, but Chen teaches:

**wherein the step of receiving a request includes receiving a
request regarding whether a first element of said,**

(See Chen fig. 5b and para 70, discloses the lists of sample EDI validation table that is also stored in a database table 425. The ELEMENT ID column at 520 matches those rows or records in the EDI map table, which have the same value in the CODE column at 502, whose CHOICE column at 506 has a value "element", and whose TYPE column at 508 has a value "ID". The CODE column at 522 in the validation table shows the shorthand terms that can appear as data contents. The VALUE column at 524 shows the long descriptions. For example, in FIG. 5A, the first element of the N1 segment (515) has a code value "98" (516), choice value "element" (517), and type value "ID" (518). It matches the four entries at 530, 531, 532, and 533 that have the same ELEMENT ID "98" in FIG. 5B. The four possible CODE values are "BT" (540), "BY" (541), "SE" (542), and "ST" (543), representing different roles or parties, "Bill-To-Party" (550), "Buying Party" (551), "Selling Party" (552), and "Ship To" (553).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Fry's XML validating parser, to include a means of receiving a request includes receiving a request regarding whether a first element of said as taught by Chen. One of ordinary skill in the art would have been motivated to perform such a modification, because Fry and Chen are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and allowed for automatic retrieval of data from a relational database

Art Unit: 2176

into an XML document using an annotated Document Type Definition ("DTD"), and would be usable by other entities (See Chen para 15).

Regarding **claim 15**, Fry teaches:

wherein the step of receiving a request includes receiving a request regarding what data type definition is associated with first element of said XML-based input stream, wherein said data type is defined in information that dictates the structure of corresponding XML data.

(See Fry para 23, discloses given namespace and element name user can get the appropriate Java classes at runtime in a mode such as generated mode. Given the stream and class, a user can get the populated java instance. The user can also have control over the deserialization. For a Java to XML case, the primitives can be mapped to XSD types. The user can also specify the XSD type. The system can convert the java instance to an XML stream or to a tree, such as a DOM tree.)

Art Unit: 2176

Regarding **claim 16**, the rejection of claim 15 is fully incorporated.

In addition, Fry teaches:

data type definition is associated with an attribute of said first element, wherein said data type that is associated with said attribute is defined in said information that dictates the structure of corresponding XML data.

(See Fry para 23, discloses the user can also specify the XSD type. The system can convert the java instance to an XML stream or to a tree, such as a DOM tree.)

Regarding **claim 17**, the rejection of claim 15-16 are fully incorporated.

In addition, Fry teaches:

data type definition in information that dictates the structure of corresponding XML data.

(See Fry para 23, discloses the user can also specify the XSD type. The system can convert the java instance to an XML stream or to a tree, such as a DOM tree.)

Regarding **claim 18**, In addition, Fry does not explicitly teach, but Chen teaches:

wherein the step of receiving a request includes receiving a request regarding a first annotation that is associated with first element of said XML-based input stream,

(See Chen FIG. 8B shows an intermediate data stream in name/value pair format.

Also, see Chen fig. 5b and para 70, discloses the lists of sample EDI validation table that is also stored in a database table 425. The ELEMENT ID column at 520

Art Unit: 2176

matches those rows or records in the EDI map table, which have the same value in the CODE column at 502, whose CHOICE column at 506 has a value "element", and whose TYPE column at 508 has a value "ID". The CODE column at 522 in the validation table shows the shorthand terms that can appear as data contents. The VALUE column at 524 shows the long descriptions. For example, in FIG. 5A, the first element of the N1 segment (515) has a code value "98" (516), choice value "element" (517), and type value "ID" (518). It matches the four entries at 530, 531, 532, and 533 that have the same ELEMENT ID "98" in FIG. 5B. The four possible CODE values are "BT" (540), "BY" (541), "SE" (542), and "ST" (543), representing different roles or parties, "Bill-To-Party" (550), "Buying Party" (551), "Selling Party" (552), and "Ship To" (553).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Fry's XML validating parser, to include a means of receiving a request includes receiving a request regarding a first annotation that is associated with first element of said XML-based input stream as taught by Chen. One of ordinary skill in the art would have been motivated to perform such a modification, because Fry and Chen are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and allowed for automatic retrieval of data from a relational database into an XML document using an annotated Document Type Definition ("DTD"), and would be usable by other entities (See Chen para 15).

Regarding **claim 19**, In addition, Fry does not explicitly teach, but Chen teaches:

wherein said information that dictates the structure of corresponding XML data comprises a second annotation definition that is associated with a second element of said XML-based input stream, and wherein the step of receiving a request includes receiving a request regarding said second annotation, the method further comprising the computer-implemented step of: before responding to said request regarding said second annotation, responding to a request regarding whether said first element is defined in said information that dictates the structure of corresponding XML data.

(See Chen FIG. 8B shows an intermediate data stream in name/value pair format.

Also, see Chen fig. 5b and para 70, discloses the lists of sample EDI validation table that is also stored in a database table 425. The ELEMENT ID column at 520 matches those rows or records in the EDI map table, which have the same value in the CODE column at 502, whose CHOICE column at 506 has a value "element", and whose TYPE column at 508 has a value "ID". The CODE column at 522 in the validation table shows the shorthand terms that can appear as data contents. The VALUE column at 524 shows the long descriptions. For example, in FIG. 5A, the first element of the N1 segment (515) has a code value "98" (516), choice value "element" (517), and type value "ID" (518). It matches the four entries at 530, 531, 532, and 533 that have the same ELEMENT ID "98" in FIG. 5B. The four possible CODE values are "BT" (540), "BY" (541), "SE" (542), and "ST" (543), representing different roles or parties, "Bill-To-Party" (550), "Buying Party" (551), "Selling Party" (552), and "Ship To" (553).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Fry's XML validating parser, to include a means of said information that dictates the structure of corresponding XML data comprises a second annotation definition that is associated with a second element of said XML-based input stream, and wherein the step of receiving a request includes receiving a request regarding said second annotation, the method further comprising the computer-implemented step of: before responding to said request regarding said second annotation, responding to a request regarding whether said first element is defined in said information that dictates the structure of corresponding XML data as taught by Chen. One of ordinary skill in the art would have been motivated to perform such a modification, because Fry and Chen are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and allowed for automatic retrieval of data from a relational database into an XML document using an annotated Document Type Definition ("DTD"), and would be usable by other entities (See Chen para 15).

Regarding **claim 20**, Fry teaches:

wherein the step of receiving a request includes receiving a request regarding a status of said validation operation with respect to a first element of said XML-based input stream.

(See Fry para 23, discloses given namespace and element name user can get the appropriate Java classes at runtime in a mode such as generated mode. Given the stream and class, a user can get the populated java instance. The user can also have control over

Art Unit: 2176

the deserialization. For a Java to XML case, the primitives can be mapped to XSD types. The user can also specify the XSD type. The system can convert the java instance to an XML stream or to a tree, such as a DOM tree.)

Regarding **claim 21**, Fry teaches:

wherein the step of receiving a request includes receiving a request via an application program interface through which information about said validation operation can be requested by an external application.

(see Fry para 23, discloses web services, XML binding, Web Service Definition Language (WSDL) code from an existing remote Java interface. A schema can be generated for complex Java data types and can also be mapped to XML Schema Definition language (XSD) types. XSD is an XML-based grammar that can be used to describe the structure of an XML document. A schema-aware validating parser can validate an XML document against an XSD schema and can report any discrepancies.

Also, see Fry para 9-1-1, describes SOAP (Simple Object Access Protocol) , SOAP is a lightweight XML protocol, which can provide both synchronous and asynchronous mechanisms for sending requests between applications.)

Art Unit: 2176

Regarding **claim 22**, Fry teaches:

wherein the step of receiving a request includes receiving a request from an event handler sent in response to an event received in a parser output stream.

(see Fry para 23, discloses web services, XML binding, Web Service Definition Language (WSDL) code from an existing remote Java interface. A schema can be generated for complex Java data types and can also be mapped to XML Schema Definition language (XSD) types. XSD is an XML-based grammar that can be used to describe the structure of an XML document. A schema-aware validating parser can validate an XML document against an XSD schema and can report any discrepancies.

Also, see Fry para 9-11, describes SOAP (Simple Object Access Protocol) , SOAP is a lightweight XML protocol, which can provide both synchronous and asynchronous mechanisms for sending requests between applications. Using the broadest reasonable interpretation, the examiner equates the claimed **an event handler** as equivalent to SOAP as taught by Fry.)

Regarding **claim 23**, the rejection of claim 13 is fully incorporated.

Art Unit: 2176

Regarding **claim 24**, Fry does not explicitly teach, but Chen teaches:

**parsing said XML-based input stream only once for both of
said validation operation and operations that are dictated by
annotations associated with elements in said XML-based input
stream.**

(See Chen fig. 8a-b, shows an intermediate data stream in name/value pair format.

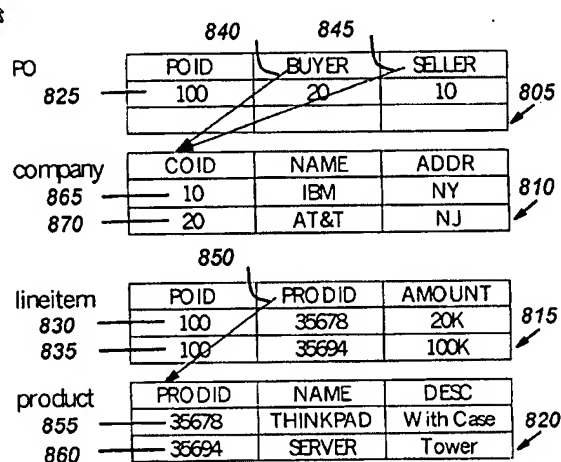


FIG. 8A

Also, see Chen para 67, teaching the three preparation stages for retrieval from a database in accordance with the preferred embodiment. The first stage 310 populates the EDI structure map or information into relational database tables. The second stage 320 creates annotations that can map the real data (from multiple data sources) to an intermediate data stream (can be a list of name/value pairs or an intermediate XML document), one mapping for each business document type. The third stage 330 annotates the target interoperable DTD that is fixed for all of the different business document types.

Also, see Chen fig. 5b and para 70, discloses the lists of sample EDI validation table that is also stored in a database table 425. The ELEMENT ID column at 520

Art Unit: 2176

matches those rows or records in the EDI map table, which have the same value in the CODE column at 502, whose CHOICE column at 506 has a value "element", and whose TYPE column at 508 has a value "ID". The CODE column at 522 in the validation table shows the shorthand terms that can appear as data contents. The VALUE column at 524 shows the long descriptions. For example, in FIG. 5A, the first element of the N1 segment (515) has a code value "98" (516), choice value "element" (517), and type value "ID" (518). It matches the four entries at 530, 531, 532, and 533 that have the same ELEMENT ID "98" in FIG. 5B. The four possible CODE values are "BT" (540), "BY" (541), "SE" (542), and "ST" (543), representing different roles or parties, "Bill-To-Party" (550), "Buying Party" (551), "Selling Party" (552), and "Ship To" (553).

	520 ELEMENT ID	522 CODE	524 VALUE
	143	810	X12 Invoice
	143	850	X12 Purchase Order
530	98	BT	Bill-To-Party
		540	550
531	98	BY	Buying Party
		541	551
532	98	SE	Selling Party
		542	552
533	98	ST	Ship To
		543	553
	66	1	D-U-N-S Number, Dun & Bradstreet
	66	8	Vendor UPC Code
	66	9	D-U-N-S+4, D-U-N-S Number with 4 Character Suffix
	66	92	Assigned by Buyer or Buyer's Agent

FIG. 5B

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Fry's XML validating parser, to include a means of parsing said XML-based input stream only once for both of said validation operation and operations that are dictated by annotations associated with elements in said XML-based input stream as taught by Chen. One of ordinary skill in the art would have been motivated to perform such a modification, because Fry and Chen are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and allowed for automatic retrieval of data from a relational database into an XML document using an annotated Document Type Definition ("DTD"), and would be usable by other entities (See Chen para 15).

Regarding **claim 25**, Fry teaches:

**wherein information that dictates the structure of
corresponding XML data in said XML-based input stream, with
which said input stream is validated in said validation operation,
comprises a plurality of schema definitions that are associated with a
plurality of corresponding XML documents that could be constituent
to said XML-based input stream**

(see Fry para 23, discloses web services, XML binding, Web Service Definition Language (WSDL) code from an existing remote Java interface. A schema can be generated for complex Java data types and can also be mapped to XML Schema Definition language (XSD) types. XSD is an XML-based grammar that can be used to

describe the structure of an XML document. A schema-aware validating parser can validate an XML document against an XSD schema and can report any discrepancies.

Also, see Fry para 9-11, describes SOAP (Simple Object Access Protocol) , SOAP is a lightweight XML protocol, which can provide both synchronous and asynchronous mechanisms for sending requests between applications.)

Regarding claims 26-38 respectively:

are directed to a computer-readable medium carrying one or more sequences of instructions, when executed by one or more processors, causes the one or more processors to perform the method recited in Claims 13-25 respectively and are similarly rejected along the same rationale.

Regarding claim 40: the rejection of claims 1 and 13 are fully incorporated.

Regarding claim 41-42: the rejection of claims 13-14 are fully incorporated.

In addition, Fry teaches:

Meta data that corresponds to said XML-based input stream.

see Fry para 23, discloses web services, XML binding, Web Service Definition Language (WSDL) code from an existing remote Java interface. A schema can be generated for complex Java data types and can also be mapped to XML Schema Definition language (XSD) types. XSD is an XML-based grammar that can be used to describe the structure of an XML document. A schema-aware validating parser can validate an XML document

Art Unit: 2176

against an XSD schema and can report any discrepancies. It is noted XML data is Meta data.

Regarding **claim 43**, Fry teaches:

**reading said annotations from an XML schema that
corresponds to said XML-based input stream.**

see Fry para 23, discloses web services, XML binding, Web Service Definition Language (WSDL) code from an existing remote Java interface. A schema can be generated for complex Java data types and can also be mapped to XML Schema Definition language (XSD) types. XSD is an XML-based grammar that can be used to describe the structure of an XML document. A schema-aware validating parser can validate an XML document against an XSD schema and can report any discrepancies.

Regarding **claim 44**, Fry teaches:

**wherein the step of causing said XML processor to generate
one or more messages includes causing said XML processor to.**

(See Fry para 27, discloses an input XML schema 100 , when unmarshalling (unmarshalling converts from a stream to an object- see Fry para 14), binding can be used by feeding an XML stream for an instance of an XML document to generated classes.

Also, see Fry para 23, discloses validating parser can validate an XML document against an XSD schema and can report any discrepancies. Using the broadest reasonable interpretation, the examiner reads the claimed **an XML processor** as equivalent to

Art Unit: 2176

validating parser can validate an XML document against an XSD schema and can report any discrepancies as taught by Fry.

Fry does not explicitly teach, but Chen teaches:

generate one or more messages that indicate how to conform said specific elements to one or more requirements of an application that uses said specific elements.

(See Chen fig. 8a-b, shows an intermediate data stream in name/value pair format.

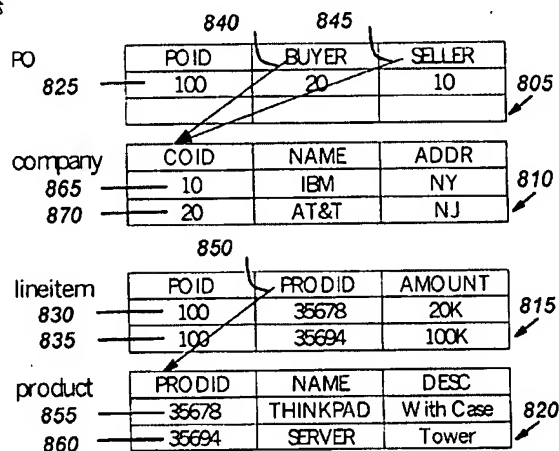


FIG. 8A

Also, see Chen para 67, teaching the three preparation stages for retrieval from a database in accordance with the preferred embodiment. The first stage 310 populates the EDI structure map or information into relational database tables. The second stage 320 creates annotations that can map the real data (from multiple data sources) to an intermediate data stream (can be a list of name/value pairs or an intermediate XML document), one mapping for each business document type. The third stage 330 annotates the target interoperable DTD that is fixed for all of the different business document types.

Art Unit: 2176

Also, see Chen fig. 5b and para 70, discloses the lists of sample EDI validation table that is also stored in a database table 425. The ELEMENT ID column at 520 matches those rows or records in the EDI map table, which have the same value in the CODE column at 502, whose CHOICE column at 506 has a value "element", and whose TYPE column at 508 has a value "ID". The CODE column at 522 in the validation table shows the shorthand terms that can appear as data contents. The VALUE column at 524 shows the long descriptions. For example, in FIG. 5A, the first element of the N1 segment (515) has a code value "98" (516), choice value "element" (517), and type value "ID" (518). It matches the four entries at 530, 531, 532, and 533 that have the same ELEMENT ID "98" in FIG. 5B. The four possible CODE values are "BT" (540), "BY" (541), "SE" (542), and "ST" (543), representing different roles or parties, "Bill-To-Party" (550), "Buying Party" (551), "Selling Party" (552), and "Ship To" (553).

520 ELEMENT ID	522 CODE	524 VALUE
143	810	X12 Invoice
143	850	X12 Purchase Order
530 98	BT 540	Bill-To-Party 550
531 98	BY 541	Buying Party 551
532 98	SE 542	Selling Party 552
533 98	ST 543	Ship To 553
66	1	D-U-N-S Number, Dun & Bradstreet
66	8	Vendor UPC Code
66	9	D-U-N-S+4, D-U-N-S Number with 4 Character Suffix
66	92	Assigned by Buyer or Buyer's Agent

FIG. 5B

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Fry's XML validating parser, to include a means of generate one or more messages that indicate how to conform said specific elements to one or more requirements of an application that uses said specific elements as taught by Chen. One of ordinary skill in the art would have been motivated to perform such a modification, because Fry and Chen are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and allowed for automatic retrieval of data from a relational database into an XML document using an annotated Document Type Definition ("DTD"), and would be usable by other entities (See Chen para 15).

Regarding **claims 45-47 respectively:**

are directed to a computer-readable medium carrying one or more sequences of instructions, when executed by one or more processors, causes the one or more processors to perform the method recited in Claims 42-44 respectively and are similarly rejected along the same rationale.

6. It is noted that any citations to specific, pages, columns, lines, or figures in the prior art references and any interpretation of the references should not be considered to be limiting in any way. A reference is relevant for all it contains and may be relied upon for all that it would have reasonably suggested to one having ordinary skill in the art. See, MPEP 2123.

Art Unit: 2176

Response to Arguments


7. The Remarks filed on 11-29-2006 has been fully considered but are moot but in view of the new ground(s) of rejection. This office action is a Non-Final Rejection in order to give the applicant sufficient opportunity to response to the new line of rejection (see rejection for details).


Conclusion

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Quoc A. Tran whose telephone number is 571-272-8664. The examiner can normally be reached on Monday through Friday from 9 AM to 5 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Herndon R. Heather can be reached on 571-272-4136. The fax phone number for the organization where this application or proceeding is assigned is (571)-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).


Quoc A. Tran
May 26, 2007


Doug Hutton
Primary Examiner
Technology Center 2100